Claims:

What is claimed is:

1. A computerized method of workload balancing for improved availability within a multitude of applications-servers and a multitude of application-clients interconnected with said application-servers by a communication network, said method comprising

caching within an application-client, availability data of a subset of currently active application-servers as potential target application-servers, and

selecting by execution of an application-request by said application-client, an application-server from said subset based on a load-balancing decision of said application-client as target application-server and is sending said application-request to said target application-server.

2. A method of workload balancing according to claim 1 further comprising,

retrieving by said application-client, said subset of currently active application-servers from a cluster database, said cluster database storing availability data of currently active application-servers.

3. A method of workload balancing according to claim 2 further comprising,

choosing said subset of currently active application-servers by approximating an even distribution of said application-servers with respect to other subsets of other application-clients.

4. A method of workload balancing according to claim 2 further comprising,

retrieving said subset of currently active application-servers by issuing a retrieve request to a server-monitor, said server-monitor monitoring said application-servers activity status.

5. A method of workload balancing according to claim 4 further comprising,

choosing by said server-monitor, said subset of currently active application-servers from said cluster database by a random selection.

6. A method of workload balancing according to claim 4 further comprising,

choosing by said server-monitor, said subset of currently active application-servers from said cluster database based on a load-balancing decision of said server-monitor.

7. A method of workload balancing according to claim 6 further comprising,

choosing by said server-monitor, said subset of currently active application-servers by approximating an even distribution of said application-servers with respect to other subsets of other application-clients.

8. A method of workload balancing according to claim 4 further comprising,

sending by said application-client together with said retrieve request, an indication of the expected amount of workload to be generated by said application-client, and

basing said server-monitor's load-balancing decision on said indication of the expected amount of workload and/or said application-servers processing power and/or said application-servers work load.

9. A method of workload balancing according to claim 1,

wherein selecting by said application-client comprises selecting said target application-server randomly.

10. A method of workload balancing according to claim 1,

wherein said availability data of said subset of currently active application-servers comprises said application-servers processing power and in addition said application-servers work load and/or said application-servers observed response time and wherein selecting by said application-client comprises selecting said target application-server approximating an even distribution of workload within said subset of currently active application-servers.

11. A method of workload balancing according to claim 1 further comprising,

dynamically updating said availability data of said subset of currently active application-servers by explicitly requesting from said server-monitor an update of said subset of currently active application-servers, and/or

dynamically updating by said server-monitor, said availability data of said subset of currently active application-servers by receiving an update of said subset of currently active application-servers.

12. A method of workload balancing according to claim 11,

wherein said explicitly requesting from said server-monitor an update of said subset of currently active application-servers occurs at one of:

prior to an application-request, or

after sending an application-request, or

periodically, or

randomly.

13. A method of workload balancing according to claim 11,

wherein, in the event an additional application-server has become active or if an active application-server has become inactive, said server-monitor takes the initiative of sending said update of said subset of currently active application-servers.

14. A method of workload balancing according to claim 13,

wherein said server-monitor sending said update of said subset of currently active application-servers comprises a lazy-multi-cast including;

iteratively selecting with a time delay, a collection of application-clients for said multitude of application-clients, and

sending each application-client of said collection an update of said subse of currently active application-servers.

15. A method of workload balancing according to claim 4,

wherein said application-servers each comprise a hot pool of one or a multitude of application-instances, said application-servers being executed in the same or a multitude of address-spaces, and

wherein for each of said hot pools a watchdog is monitoring said hot pool's activity status, and

wherein each of said watchdogs is informing said server-monitor on the workload of said monitored application-servers and its current activity status.

16. A method of workload balancing according to claim 15,

wherein at least one of said watchdogs is monitoring the activity status of other watchdogs and, in the event that said one watchdog detects the unavailability of any one of said other watchdogs, said one watchdog informs said server-monitor of said corresponding activity status and said server-monitor updates said cluster database accordingly.

17. An apparatus for workload balancing giving improved availability within a multitude of applications-servers and a multitude of application-clients interconnected with said application-servers by a communication network, said apparatus comprising

means for caching within an application-client, availability data of a subset of currently active application-servers as potential target application-servers, and

means for selecting by execution of an application-request by said application-client, an application-server from said subset based on a load-balancing decision of said application-client as target application-server and is sending said application-request to said target application-server.

18. An apparatus for workload balancing according to claim 17 further comprising,

means for retrieving by said application-client, said subset of currently active application-servers from a cluster database, said cluster database storing availability data of currently active application-servers.

19. An apparatus for workload balancing according to claim 18 further comprising,

means for choosing said subset of currently active application-servers by approximating an even distribution of said application-servers with respect to other subsets of other application-clients.

20. An apparatus for workload balancing according to claim 18 further comprising,

means for retrieving said subset of currently active application-servers by issuing a retrieve request to a server-monitor, said server-monitor monitoring said application-servers activity status.

21. An apparatus for workload balancing according to claim 20 further comprising,

means for choosing by said server-monitor, said subset of currently active application-servers from said cluster database by a random selection.

22. An apparatus for workload balancing according to claim 20,

means for choosing by said server-monitor, said subset of currently active application-servers from said cluster database based on a load-balancing decision of said server-monitor.

23. An apparatus for workload balancing according to claim 22 further comprising,

means for choosing by said server-monitor, said subset of currently active application-servers by approximating an even distribution of said application-servers with respect to other subsets of other application-clients.

24. An apparatus for workload balancing according to claim 20 further comprising,

means for sending by said application-client together with said retrieve request, an indication of the expected amount of workload to be generated by said application-client, and

means for basing said server-monitor's load-balancing decision on said indication of the expected amount of workload and/or said application-servers processing power and/or said application-servers work load.

25. An apparatus for workload balancing according to claim 17,

wherein said means for selecting by said application-client comprises means for selecting said target application-server randomly.

26. An apparatus for workload balancing according to claim 17,

wherein said availability data of said subset of currently active application-servers comprises said application-servers processing power and in addition said application-servers work load and/or said application-servers observed response time and wherein means for selecting by said application-client comprises means for selecting said target application-server approximating an even distribution of workload within said subset of currently active application-servers.

27. An apparatus for workload balancing according to claim 17 further comprising,

means for dynamically updating said availability data of said subset of currently active application-servers by explicitly requesting from said server-monitor an update of said subset of currently active application-servers, and/or

means for dynamically updating by said server-monitor, said availability data of said subset of currently active application-servers by receiving an update of said subset of currently active application-servers.

28. An apparatus for workload balancing according to claim 27,

wherein said explicitly requesting from said server-monitor an update of said subset of currently active application-servers occurs at one of:

prior to an application-request, or

after sending an application-request, or

periodically, or

randomly.

29. An apparatus for workload balancing according to claim 27,

wherein, in the event an additional application-server has become active or if an active application-server has become inactive, said server-monitor takes the initiative of sending said update of said subset of currently active application-servers.

30. An apparatus for workload balancing according to claim 29,

wherein said server-monitor sending said update of said subset of currently active application-servers comprises a lazy-multi-cast including;

means for iteratively selecting with a time delay, a collection of application-clients for said multitude of application-clients, and

means for sending each application-client of said collection an update of said subse of currently active application-servers.

31. An apparatus for workload balancing according to claim 20,

wherein said application-servers each comprise a hot pool of one or a multitude of application-instances, said application-servers being executed in the same or a multitude of address-spaces, and

wherein for each of said hot pools a watchdog is monitoring said hot pool's activity status, and

wherein each of said watchdogs is informing said server-monitor on the workload of said monitored application-servers and its current activity status.

32. An apparatus for workload balancing according to claim 31,

wherein at least one of said watchdogs is monitoring the activity status of other watchdogs and, in the event that said one watchdog detects the unavailability of any one of said other watchdogs, said one watchdog informs said server-monitor of said corresponding activity status and said server-monitor updates said cluster database accordingly.

33. A computer program product comprising a computer usable medium having computer readble program code means therein for workload balancing giving improved availability within a multitude of applications-servers and a multitude of application-clients interconnected with said application-servers by a communication network, said computer program product comprising

computer readable program code means for caching within an application-client, availability data of a subset of currently active application-servers as potential target application-servers, and

computer readable program code means for selecting by execution of an application-request by said application-client, an application-server from said subset based on a load-balancing decision of said application-client as target application-server and is sending said application-request to said target application-server.

34. The computer program product for workload balancing according to claim 33 further comprising,

computer readable program code means for retrieving by said application-client, said subset of currently active application-servers from a cluster database, said cluster database storing availability data of currently active application-servers.

35. The computer program product for workload balancing according to claim 34 further comprising,

computer readable program code means for choosing said subset of currently active application-servers by approximating an even distribution of said application-servers with respect to other subsets of other application-clients.

36. The computer program product for workload balancing according to claim 34 further comprising,

computer readable program code means for retrieving said subset of currently active application-servers by issuing a retrieve request to a server-monitor, said server-monitor monitoring said application-servers activity status.

37. The computer program product for workload balancing according to claim 36 further comprising,

computer readable program code means for choosing by said server-monitor, said subset of currently active application-servers from said cluster database by a random selection.

38. The computer program product for workload balancing according to claim 36,

computer readable program code means for choosing by said server-monitor, said subset of currently active application-servers from said cluster database based on a load-balancing decision of said server-monitor.

39. The computer program product for workload balancing according to claim 38 further comprising,

computer readable program code means for choosing by said server-monitor, said subset of currently active application-servers by approximating an even distribution of said application-servers with respect to other subsets of other application-clients.

40. The computer program product for workload balancing according to claim 36 further comprising,

computer readable program code means for sending by said application-client together with said retrieve request, an indication of the expected amount of workload to be generated by said application-client, and

computer readable program code means for basing said server-monitor's load-balancing decision on said indication of the expected amount of workload and/or said application-servers processing power and/or said application-servers work load.

41. The computer program product for workload balancing according to claim 33,

wherein said computer readable program code means for selecting by said application-client comprises computer readable program code means for selecting said target application-server randomly.

42. The computer program product for workload balancing according to claim 33,

wherein said availability data of said subset of currently active application-servers comprises said application-servers processing power and in addition said application-servers work load and/or said application-servers observed response time and

wherein computer readable program code means for selecting by said application-client comprises computer readable program code means for selecting said target application-server approximating an even distribution of workload within said subset of currently active application-servers.

43. The computer program product for workload balancing according to claim 33 further comprising,

computer readable program code means for dynamically updating said availability data of said subset of currently active application-servers by explicitly requesting from said server-monitor an update of said subset of currently active application-servers, and/or

computer readable program code means for dynamically updating by said server-monitor, said availability data of said subset of currently active application-servers by receiving an update of said subset of currently active application-servers.

44. The computer program product for workload balancing according to claim 43,

wherein said explicitly requesting from said server-monitor an update of said subset of currently active application-servers occurs at one of:

prior to an application-request, or
after sending an application-request, or
periodically, or
randomly.

45. The computer program product for workload balancing according to claim 43,

wherein, in the event an additional application-server has become active or if an active application-server has become inactive, said server-monitor takes the initiative of sending said update of said subset of currently active application-servers.

46. The computer program product for workload balancing according to claim 45,

wherein said server-monitor sending said update of said subset of currently active application-servers comprises a lazy-multi-cast including;

computer readable program code means for iteratively selecting with a time delay, a collection of application-clients for said multitude of application-clients, and

computer readable program code means for sending each application-client of said collection an update of said subse of currently active application-servers.

47. The computer program product for workload balancing according to claim 36,

wherein said application-servers each comprise a hot pool of one or a multitude of application-instances, said application-servers being executed in the same or a multitude of address-spaces, and

wherein for each of said hot pools a watchdog is monitoring said hot pool's activity status, and

wherein each of said watchdogs is informing said server-monitor on the workload of said monitored application-servers and its current activity status.

48. The computer program product for workload balancing according to claim 47,

wherein at least one of said watchdogs is monitoring the activity status of other watchdogs and, in the event that said one watchdog detects the unavailability of any one of said other watchdogs, said one watchdog informs said server-monitor of said corresponding activity status and said server-monitor updates said cluster database accordingly.